

Ασκήσεις στις Συναρτήσεις

1. Να αναπτύξετε συνάρτηση `double abs(double n)` που επιστρέφει την απόλυτη τιμή του `n`
2. Να αναπτύξετε συνάρτηση που υπολογίζει και τυπώνει την τιμή της $f(x)=x^4+5x^3+2x^2+8x+1$ για τις τιμές του `x=1..100`, όπου `x` ακέραιος.
3. Να αναπτύξετε συνάρτηση που επιστρέφει την δύναμη του `b` υψωμένο στην `e`, με `b` πραγματικός και `e` ακέραιος.
4. Να αναπτυχθεί συνάρτηση που λαμβάνει 3 παραμέτρους που αντιπροσωπεύουν μια εξίσωση 2^{ου} βαθμού ($ax^2+bx+c=0$) και τυπώνει την λύση της εφόσον υπάρχει ή διαφορετικά το μήνυμα `δεν υπάρχει λύση, διακρίνουσα <0`.
5. Να αναπτύξετε συνάρτηση που τυπώνει τους `n` πρώτους όρους της ακολουθίας Fibonacci (https://en.wikipedia.org/wiki/Fibonacci_number)
6. Να αναπτυχθεί συνάρτηση που ταξινομεί έναν πίνακα ακεραίων κατά αύξουσα σειρά.
7. Να αναπτυχθεί συνάρτηση που επιστρέφει `true` αν η ακέραιη παράμετρός της είναι πρώτος αριθμός.
8. Να αναπτυχθεί συνάρτηση που τυπώνει τους πρώτους αριθμούς από το 1 έως το 100.
9. Να αναπτυχθεί συνάρτηση που τυπώνει τους αριθμούς από το 1 έως το 100 που δεν είναι πρώτοι.
10. Να αναπτυχθεί συνάρτηση που λαμβάνει ένα μονοδιάστατο πίνακα χαρακτήρων και επιστρέφει έναν αντίγραφο του ανεστραμμένο.
11. Να αναπτυχθεί συνάρτηση που λαμβάνει μια `boolean` παράμετρο και επιστρέφει έναν πίνακα με τους χαρακτήρες της Αγγλικής, κεφαλαίους ή πεζούς, ανάλογα με την τιμή της παραμέτρου.
12. Να υλοποιηθεί Java application που λαμβάνει 3 παραμέτρους που αντιπροσωπεύουν μια εξίσωση 2^{ου} βαθμού ($ax^2+bx+c=0$) και τυπώνει την λύση της εφόσον υπάρχει ή διαφορετικά το μήνυμα `δεν υπάρχει λύση, διακρίνουσα <0`.
13. Να αναπτυχθεί συνάρτηση που επιστρέφει την δύναμη ενός πραγματικού αριθμού υψωμένου σε έναν εκθέτη. Θεωρήστε πως ο εκθέτης είναι ακέραιος, μη αρνητικός.
14. Να αναπτυχθεί συνάρτηση που τυπώνει τους όρους της ακολουθίας Fibonacci. Ο τελευταίος όρος που εκτυπώνεται καθορίζεται από κατάλληλη παράμετρο της συνάρτησης.
15. Να αναπτυχθεί η κλάση `Tables` που περιλαμβάνει τις ακόλουθες συναρτήσεις

```
static int binarySearch(int[] tbl, int schElement)
```

Binary search in a one dimension array of integers to find a specific value

```
static int cntOccurances(int[] t, int sE)
```

Counts the number of occurances of a specific value in an one dimension array of integers

```
static boolean equals(int[] tA, int[] tB)
```

Compares for equality between two (one-dimensional) arrays of integers.

```
static boolean equals(java.lang.String[] tA, java.lang.String[] tB)
```

Compares for equality between two (one-dimensional) arrays of Strings.

```
static int    max(int[] t)
```

Search for the max element in an integer array

```
static void   print(int[] array)
```

Prints the contents of a one dimension array of chars in one line seperated each other by a space character

```
static void   println (int[] array)
```

Prints the contents of a one dimension array of chars in one line seperated each other by a space character.

```
static int[]  reverse(int[] input)
```

Produces a reversed array

```
static int    seqSearch(int[] t, int sE)
```

Sequential search in an one dimension array of integers to find a specific value

```
static int    seqSearch(java.lang.String[] t, java.lang.String sE)
```

Sequential search in an one dimension array of Strings to find a specific value

```
static void   sort(int[] array)
```

Ταξινόμηση κατα αύξουσα σειρά ενός μονοδιάστατου πίνακα ακεραίων.

```
static int    sum(int[] t)
```

Calculates the sum of all elements of a one-dimension int array